



KGiSL INSTITUTE OF TECHNOLOGY

Coimbatore – 641035

Institution code : 7117

Traffic Management system using IBM Internet of Things

MENTOR:

MRS.INDU POORNIMA.R

TEAM MEMBER:

Davis Niranjana.j

Step 1: Defining the system requirements and objectives of the smart traffic management project

Objectives:

Improve traffic flow: Improve the efficiency of traffic flow by reducing delays by reducing congestion at major intersections and sidewalks.

Enhance safety: Improve road safety by reducing accidents and ensuring smooth traffic shifts.

Reduce Environmental Impact: Reduce greenhouse gas and fuel consumption by reducing the time vehicles spend idle in traffic.

Real-time analytics: Use real-time monitoring of traffic conditions and respond quickly to incidents.

Data-Driven Decision Making: Use data analytics to make informed decisions for traffic management and infrastructure improvement.

Increase public transit integration: Encourage the use of public transit by aligning traffic lights and roads with public transit systems.

Access Improvements: Increase accessibility for pedestrians and cyclists through improved intersections and trails.

Scalability: Ensure the system can scale to meet future developments and technological developments.

Requirements:

Traffic Sensors: Place sensors (e.g., cameras, ultrasonic sensors, pressure sensors) at key locations to collect real-time traffic data

Centralized Control Center: Establish a centralized control center to manage and monitor the vehicle system.

Smart traffic lights: Use smart lights that can adapt to traffic conditions and prioritize certain traffic patterns.

Communication system: Create a reliable communication system to transfer data between sensors, control center, and traffic lights.

Data Analytics Platform: Use a data analytics platform to process and analyze incoming traffic data.

Machine Learning Algorithms: Develop machine learning algorithms to predict traffic patterns and optimize traffic light timings.

Real-Time Data Display: Create user-friendly dashboards and mobile apps for the public to access real-time traffic data.

Public Awareness Campaign: Launch a public awareness campaign to inform drivers and pedestrians about the smart traffic management system.

Traffic Light Synchronization: Ensure that traffic lights are synchronized with public transport schedules.

Scalable Architecture: Design a scalable system that can adapt to increased traffic and new technologies.

Step 2: Choosing Sensors and Cameras

Traffic sensors:

Ultrasonic sensors: Ultrasonic sensors are suitable for measuring the distance between the sensor and nearby vehicles, and providing information on traffic volume and flow speed.

Infrared sensors: Infrared sensors can detect the presence of vehicles by measuring changes in infrared radiation. They work well for controlling traffic flow.

Car recognition cameras:

Raspberry Pi Camera Module: This camera module is a cost-effective solution for recording photos and videos. It can be used for vehicle identification and license plate identification (LPR).

USB Webcams: USB webcams are easy to connect to the Raspberry Pi and provide video data that can be used for real-time monitoring.

Environmental Awareness:

Thermal properties: Thermal properties can provide environmental issues that can affect road conditions such as icy roads in winter.

Moisture content: Moisture content can help measure weather conditions and their impact on road safety.

Traffic light cameras:

These cameras are used to monitor traffic lights at intersections. They ensure that the system can adjust traffic light timing according to traffic conditions.

License Plate Recognition (LPR) Camera (Optional):

LPR cameras can be used for advanced applications such as monitoring entry and exit points, tracking traffic violators and collecting fines.

Step 3: setting up Raspberry Pi

Raspberry Pi 3 (or a model of your choice)

Power supply for the Raspberry Pi

MicroSD card with Raspbian OS (or your preferred OS)

USB peripherals (keyboard, mouse, and a monitor for initial setup)

Step 4: Developing IOT software

Python code for interfacing with sensors and cameras and sending data to Azure IoT Hub

CODE:

```
import time
import random
from azure.iot.device import IoTHubDeviceClient
CONNECTION_STRING = "
HostName=hostforiotproject.azure-devices.net;DeviceId=raspberry-pi-004;SharedAccessKey=q
ghd30aseqgh456sd"
client = IoTHubDeviceClient.create_from_connection_string(CONNECTION_STRING)
def simulate_sensor_data():
    temperature = random.uniform(10.0, 40.0)
    humidity = random.uniform(20.0, 80.0)
    return {"temperature": temperature, "humidity": humidity}
def preprocess_and_filter(data):
    if data["temperature"] < 0 or data["temperature"] > 50:
        return None
    if data["humidity"] < 0 or data["humidity"] > 100:
        return None
    return data

while True:
    try:
        sensor_data = simulate_sensor_data()
        processed_data = preprocess_and_filter(sensor_data)
        if processed_data:
            message = str(processed_data)
            client.send_message(message)
            print(f"Message sent: {message}")
            time.sleep(10)
        except Exception as e:
            print(f"Error: {str(e)}")
client.shutdown()
```

Step 5: Azure IoT Hub Setup

Signed in to Azure IOT Hub

Registered RaspberryPI as device in azure IOT Hub and received Connection string

Step 6: Azure IoT Cloud Simulator

Objective: Develop a virtual testing environment for traffic scenario simulation.

Accomplishments: Successfully set up and configured the Azure IoT Cloud Simulator, enabling the creation of virtual traffic scenarios.

Highlights: Created device templates and simulation models that accurately mimic real-world traffic behaviors, allowing for rigorous testing and validation.

Step 7: Data Transmission

Objective: Enable secure transmission of sensor data from Raspberry Pi devices to Azure IoT Hub.

Accomplishments: Modified Raspberry Pi code to send sensor data using connection strings, with robust security measures, including TLS.

Highlights: Achieved secure and reliable data transmission, ensuring that real-time traffic data reaches the Azure cloud for processing.

Step 8: Data Processing and Analysis

Objective: Implement real-time data processing and traffic analysis in the Azure cloud.

Accomplishments: Set up Azure services like Azure Stream Analytics, Azure Functions, and Azure Machine Learning for data processing.

Highlights: Implemented advanced traffic management logic, including real-time congestion detection and other features to improve traffic flow.

Step 9: Visualization and User Interface

Objective: Create user-friendly dashboards for real-time traffic data visualization.

Accomplishments: Developed web-based and mobile dashboards using Azure Web Apps and Power BI.

Highlights: The system provides a visually rich interface for both traffic operators and the public to monitor traffic conditions and receive alerts.

Step 10: Testing and Optimization

Objective: Rigorously test and optimize the entire system for performance and reliability.

Accomplishments: Successfully tested the system's components, including sensor accuracy, data transmission, cloud processing, and visualization.

Highlights: Extensive optimization efforts were made to ensure the system's performance and scalability, providing real-time traffic insights with precision.

Step 11: Deployment

Objective: Deploy the smart traffic management system in a real-world traffic environment.

Accomplishments: After thorough testing and optimization, the system has been deployed in the desired location.

Highlights: The system is now actively managing traffic, optimizing signal timings, and providing real-time traffic information to commuters, contributing to safer and more efficient transportation.

Step 12: Monitoring and Maintenance

Objective: Implement continuous monitoring and maintenance procedures to ensure system reliability.

Accomplishments: Established monitoring mechanisms for real-time traffic data and system health.

Highlights: Proactive maintenance and system adjustments are ongoing to maintain system performance and reliability, keeping urban traffic optimized and safe.